

Outsourced Storage & Proofs of Retrievability

Hovav Shacham, UC San Diego
Brent Waters, SRI International

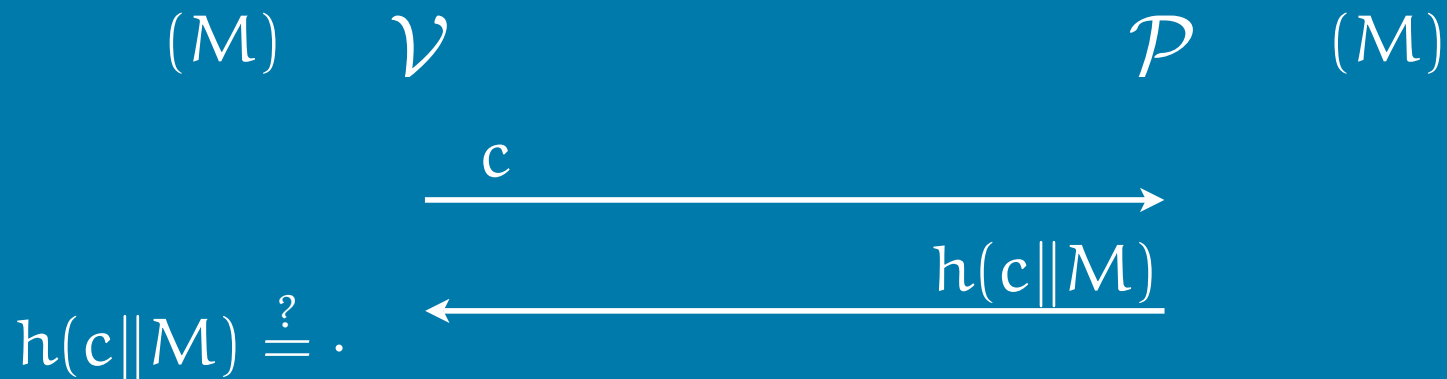
The Setting

- Client stores (long) file with server
 - Wants to be sure it's actually there
- Motivation: online backup; SaaS
- Long-term reliable storage is expensive

Example Protocols



Kotla, Alvisi, Dahlin, Usenix 2007:



How do we evaluate
protocols of this sort?

Systems Criteria

- Efficiency:
 - Storage overhead
 - Computation (including # block reads)
 - Communication
- Unlimited use
- Stateless verifiers
- Who can verify? File owner? anyone?

Crypto criterion

- Only an adversary **storing** the file can pass the verification test
- Possible to **extract** M from any prover P' via black-box access
- (Cf. ZK proof-of-knowledge)
- Insight due to Naor, Rothblum, FOCS 2005 and Juels, Kaliski, CCS 2007

Security Model — I

- **Keygen**: output secret key sk
- **Store** (sk , file M):
output tag t , encoded file M^*
- **Proof-of-storage** protocol:
$$\{0, 1\} \stackrel{\mathcal{R}}{\leftarrow} (\mathcal{V}(sk, t) \Rightarrow \mathcal{P}(t, M^*))$$
- **Public verifiability**:
 - Keygen outputs keypair (pk, sk)
 - Verifier algorithm takes only pk

Security Model — II

- Challenger generates sk
- Adversary makes queries:
 - “store M_i ” \Rightarrow get t_i, M_i^*
 - “protocol on t_i ” \Rightarrow interact w/ $V(sk, t_i)$.
- Finally, adversary outputs:
 - challenge tag t from among $\{t_i\}$
 - description of **cheating prover** P' for t

Security Model — III

- Security guarantee:

\exists **extractor** algorithm Extr st. when

$$\Pr\left[\left(\mathcal{V}(\text{sk}, t) \rightleftharpoons \mathcal{P}'\right) = 1\right] \geq \epsilon$$

we have

$$\text{Extr}(\text{sk}, t, \mathcal{P}') = M$$

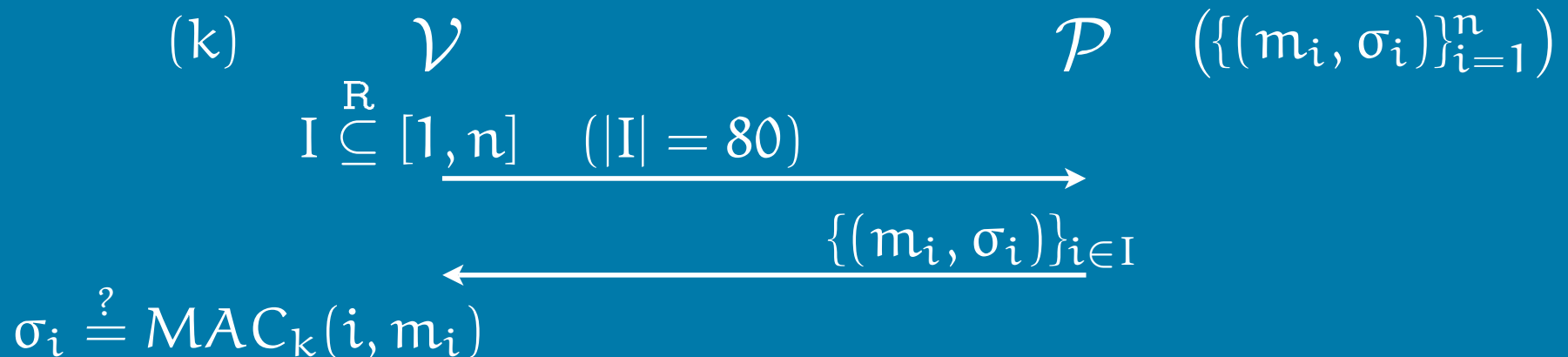
except with negligible probability

Probabilistic Sampling

- Want to check 80 blocks at random, not entire file
- $\Pr[\text{detect 1-in-}10^6 \text{ erasure}] < 0.01\%$
- $\Pr[\text{detect 50\% erasure}] = 1 - (1/2)^{80}$
- So: encode $M \Rightarrow M^*$ st. any 1/2 of blocks suffice to recover M : **erasure code**
- Due to Naor, Rothblum, FOCS 2005

The Simple Solution

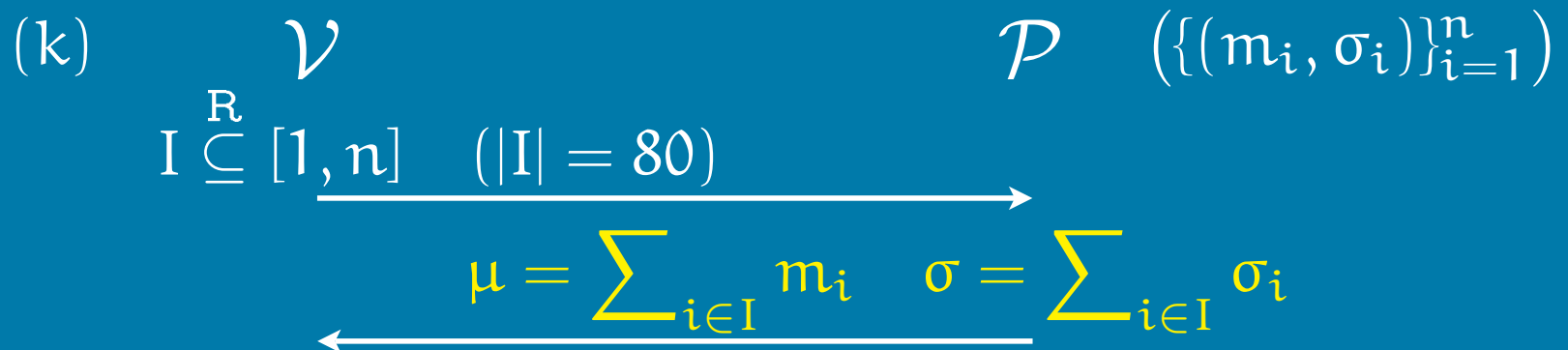
- Store:
 - erasure encode $M \Rightarrow M^*$
 - for each block m_i of M^* ,
store **authenticator** $\sigma_i = \text{MAC}_k(i, m_i)$
- Proof of storage:



Lower communication
using homomorphic
authenticators

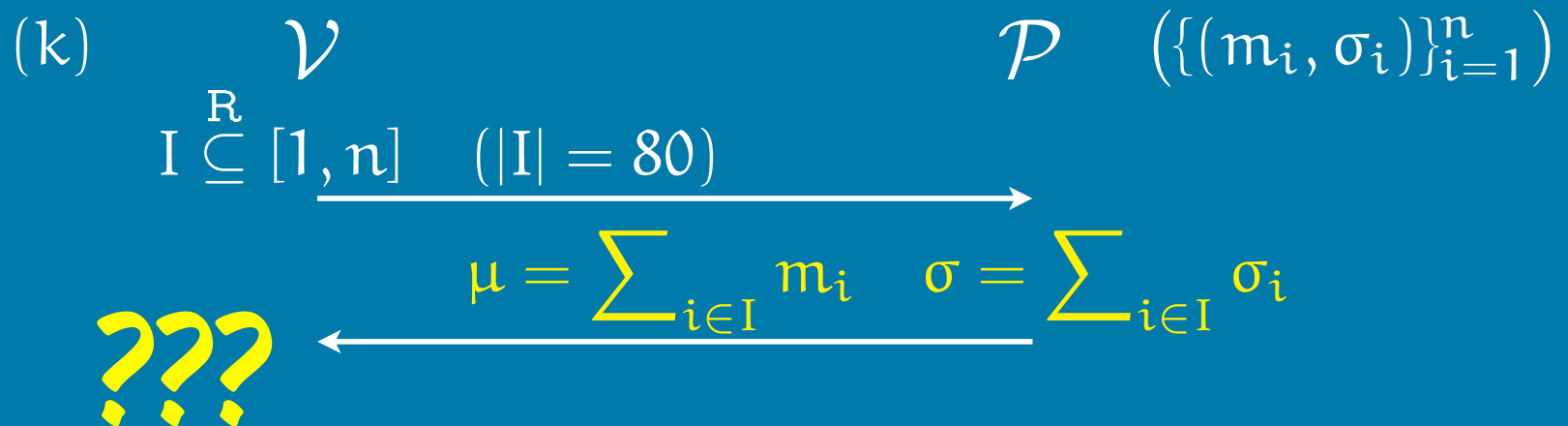
Improved Solution (Try #1)

- Downside to simple solution:
response is 80 blocks, 80 authenticators
- Let's send Σm_i instead!



Improved Solution (Try #1)

- Downside to simple solution:
response is 80 blocks, 80 authenticators
- Let's send Σm_i instead!



Homomorphic Authenticators

- Problem: have linear combination of messages m_i
- Need to authenticate via some function of $\{\sigma_i\}$
- Ateniese et al., CCS 2007:
RSA-based homomorphic authenticators;

$$\prod_i \sigma_i^{v_i} \text{ authenticates } \sum_i v_i m_i$$

Our Contributions

1. Efficient homomorphic authenticators based on PRFs and on bilinear groups
2. A full proof for (improved) simple protocol, against *arbitrary* adversaries

PRF Authenticator

- PRF $f: \{0,1\}^* \rightarrow K$; $m_i \in K$; $K: \text{GF}(2^{80})$ or \mathbb{Z}_p
- Keygen: PRF key k ; $\alpha \in K$
- Authenticate: $\sigma_i \leftarrow f_k(i) + \alpha \cdot m_i$
- Aggregate:

$$\sigma \leftarrow \sum v_i \sigma_i \quad \text{and} \quad \mu \leftarrow \sum v_i m_i$$

- Verify:

$$\sigma \stackrel{?}{=} \sum v_i f_k(i) + \alpha \mu$$

BLS Authenticator

- Bilinear map $e: G_1 \times G_2 \rightarrow G_T$, $\langle u \rangle = G_1$.
- Keygen: $sk: x \in \mathbb{Z}_p$; $pk: v = g_2^x \in G_2$.
- Authenticate: $\sigma_i \leftarrow [H(i)u^{m_i}]^x$
- Aggregate:

$$\sigma \leftarrow \prod \sigma_i^{v_i} \quad \text{and} \quad \mu \leftarrow \sum v_i m_i$$

- Verify:

$$e(\sigma, g) \stackrel{?}{=} e\left(u^\mu \cdot \prod H(i)^{v_i}, v\right)$$

Improved Solution (Try #2)

$$(k, \alpha) \quad \mathcal{V}$$

$$\mathcal{P} \quad (\{(m_i, \sigma_i)\}_{i=1}^n)$$

$$I \stackrel{R}{\subseteq} [1, n] \quad (|I| = 80)$$

$$v_i \stackrel{R}{\leftarrow} K \quad \text{for } i \in I$$

$$Q = \{(i, v_i)\}$$

$$\mu \leftarrow \sum_{(i, v_i) \in Q} v_i m_i$$

$$\sigma \leftarrow \sum_{(i, v_i) \in Q} v_i \sigma_i$$

$$\sigma \stackrel{?}{=} \sum_{(i, v_i) \in Q} v_i f_k(i) + \alpha \mu$$

μ, σ

Communication & storage

- PRF solution: 80-bit μ , 80-bit σ
- BLS solution: 160-bit μ , 160-bit σ
- But: 100% storage overhead
- Storage/communication tradeoff:
 - split each block into s sectors
 - one authenticator per block:
 - response: $(1+s) \times 80$ bits [or $\times 160$ bits]
 - storage overhead: $1/s$

The proof of security

Security Proof Outline

1. “**Straitening**”: whenever (μ, σ) verify correctly, μ was computed as $\sum v_i m_i$
2. “**Extraction**”: can extract 1/2 of blocks from prover P' that outputs $\mu = \sum v_i m_i$ on ε -fraction of queries, \perp otherwise
3. “**Decoding**”: recover M from any 1/2 of M^* blocks

Attack on Improved Solution Try #1

- Attacker picks index i^*
- For $i \neq i^*$, sets $a_i \leftarrow \pm 1$, stores $m' \leftarrow m_i + a_i m_{i^*}$
- for query l st. $i^* \notin l$, compute

$$\mu' = \sum_{i \in I} m'_i = \sum_{i \in I} (m_i + a_i m_{i^*}) = \mu + m_{i^*} \sum_{i \in I} a_i$$

- this is correct if $\#(+1) = \#(-1)$ in $\sum a_i$:

$$\Pr \left[0 = \sum_{i \in I} a_i \right] = \binom{80}{40} \cdot \frac{1}{2^{80}} \approx 8.89\%$$

Attack (cont.)

Attacker knows dim $(n-1)$ subspace:

$$\begin{pmatrix} 1 & & & \dots & 0 & \pm 1 \\ & 1 & & \ddots & \vdots & \pm 1 \\ & & \ddots & & & \pm 1 \\ \vdots & \ddots & & & 1 & \pm 1 \\ 0 & \dots & & & & 1 & \pm 1 \end{pmatrix}$$

But he doesn't know any single block!

Conclusion

- Homomorphic authenticators from PRFs, BLS
- “Improved Solution, Try #2”:
 - compact response (& query in r.o. model)
 - secure against arbitrary adversarial behavior
- Security requires proof — some okay-looking schemes are insecure

<http://cs.ucsd.edu/~hovav/papers/sw08.html>